Equations with heterogeneous coefficients

```
clear
set_demo_defaults
col = marc_colors();
```

Discretization of the heterogeneous coefficient

Consider the discretization of incompressible flow with heterogenous thermal conductivity

Continuous equation: $-\nabla \cdot [K(\mathbf{x})\nabla T] = f_s$ on $x \in [0, L]$

Discrete equation: `-D*[Kd*G]*u = fs`

Here we have introduced a new (Nx+1) by (Nx+1) matrix **`Kd`** that accounts for the heterogeneous coefficient on the cell faces.

However, the thermal conductivity is generally evaluated at the cell center and must be *averaged* to the cell faces.

The discrete equation for the flux on the i-th face is

$$q_i = -K_{i-1/2}\frac{u_i - u_{i-1}}{\Delta x}$$

where $K_{i-1/2}$ is a suitable average of K_{i-1} and K_i. This average could be either:

1. Arithmetic: $K_{i-1/2} = \dfrac{K_{i-1} + K_i}{2}$
2. Harmonic: $K_{i-1/2} = \dfrac{2}{\dfrac{1}{K_{i-1}} + \dfrac{1}{K_i}}$

These two formulas a special cases of a general power-law mean

$$\langle K \rangle_p = \left(\frac{1}{2}\left(K_1^p + K_2^p\right)\right)^{1/p}$$

where the arithmetic average is recovered for $p = 1$ and the harmonic average for $p = -1$.

Suppose we compute a Nx+1 by 1 column vector **`Kmean`** that contains these averages to the faces. then we could compute the flux in Matlab using elementwise multiplication

`q = -Kmean.*(G*u)`

While this is correct, it does not allow us to write the linerar operator `L = -D*Kd*G`.

An equivalent way to compute the fluxes is to place the vector Kman on the diagonal of the (nx+1) by (Nx+1) matrix Kd and then use matrix vector multiplication

`q = -Kd*(G*u)`

Therefore the matrix **Kd** simply contais the averages of Kfron the cells to the faces along its diagonal.

Implementation in `comp_mean.m`

The computation of these averages will be encapsulated in the function `comp_mean.m`. From the function `build_ops.m` we have the **M** matrix which "averages" values from the cell centers to the faces. This is an arithmetic average! Given M we can implement the averages as follows:

- arithmetic (p=1): **`Kmean = M*K`**
- harmonic (p=-1): **`Kmean = 1./(M*(1./K))`**
- general power mean: **`Kmean = (M*K.^p).^(1/p)`**

The resultin column vector **Kmean** is then placed on the diagonal of the Nf by Nf matrix **Kd**.

Example: Heterogeneous column - Analytic solution

Consider one-dimensional heat conduction through a core of of length L = 100 cm

$-\nabla[K(x)\nabla T] = 0$ on $x \in [0, L]$

with heterogeneous thermal conductivity

$$K(x) = \begin{cases} K_1 = 5 \times 10^{-3}, & \text{for } x < 0.2 \text{ cm} \\ K_2 = 5 \times 10^{-5}, & \text{for } 0.2 < x < 0.4 \text{ cm} \\ K_3 = 5 \times 10^{-3}, & \text{for } x > 0.4 \text{ cm} \end{cases}$$

and two Dirichlet boundary conditions, $T_L = 120$ C and $T_R = 100$ C.

```
%% Parameters
K = [5e-3 5e-5 5e-3 ];  % thermal conductivity values
l = [20 20 60];         % length of the corresponding column segments
TL = 120;               % temperature on the left
TR = 100;               % temperature on the right
```

The temperature is piecewise linear and the flux is constant. The analytic solution can be obtained computing the heads, $u_1$and u_2, at the two interfaces between the three segments. Applying Fourier's law to each segment

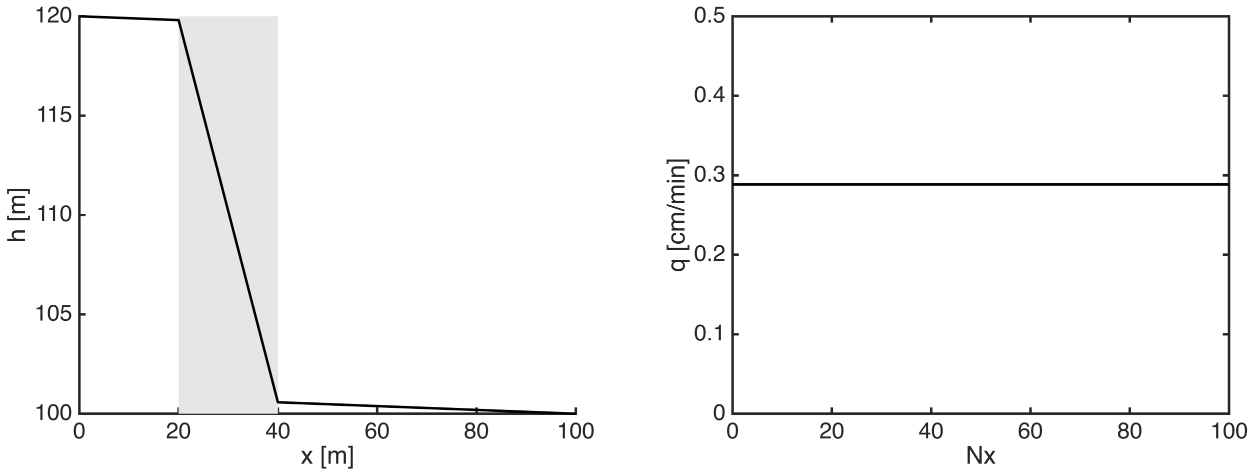
$$q = -K_1\frac{T_1 - T_L}{l_1} = -K_2\frac{T_2 - T_1}{l_2} = -K_3\frac{T_R - T_2}{l_3}$$

we have three simultaneous equations that can be solved for $u_1$and u_2.

$T_1 = \dfrac{K_1(K_2l_3 + l_2K_3)T_L + l_1K_2K_3T_R}{K_1K_2l_3 + K_1l_2K_3 + l_1K_2K_3}$ and $T_2 = \dfrac{K_3(K_1l_2 + l_1K_2)T_R + l_3K_1K_2T_L}{K_1K_2l_3 + K_1l_2K_3 + l_1K_2K_3}$

the flux can then be obtained either by computing the weighted harmonic average of $K_\perp^*$ and applying Fourier's law to the whole core, $q = -K_\perp^* \frac{T_R - T_L}{L}$ or by applying Darcy's law to any segment as shown above.

```
%% Analytic solution
denom = K(1)*K(2)*l(3) + K(1)*l(2)*K(3) + l(1)*K(2)*K(3);
T1 = (K(1)*(K(2)*l(3)+l(2)*K(3))*TL+l(1)*K(2)*K(3)*TR)/denom;
T2 = (K(3)*(K(1)*l(2)+l(1)*K(2))*TR+l(3)*K(1)*K(2)*TL)/denom;
Ta = [TL T1 T2 TR];
qa = - K(1)*(T1-TL)/l(1);
xa = [0 cumsum(l)];

figure('position',[10 10 1200 600])
subplot 121
patch([20 40 40 20],[100 100 120 120],.9*[1 1 1],'edgecolor','none'), hold
on
plot(xa,Ta,'k-','linewidth',2)
xlabel 'x [m]', ylabel 'h [m]'
pbaspect([1 .8 1])

subplot 122
% semilogy([0 Nx(end)],qa*[1 1],'k-','linewidth',2), hold on
plot([0 100],qa*[1 1]*6000,'k-','linewidth',2) % 6000 is conversion from
m/s to cm/min!
xlim([0 100]),ylim([0 0.5])
xlabel 'Nx', ylabel 'q [cm/min]'
pbaspect([1 .8 1])
```



The solution shows that most of the head drop occurs in the second segment (shaded grey) where the thermal conductivity is lower. This indicates that most of the temperature drop occurs across the low conductivity segment.

Heterogeneous column - Numerical solution

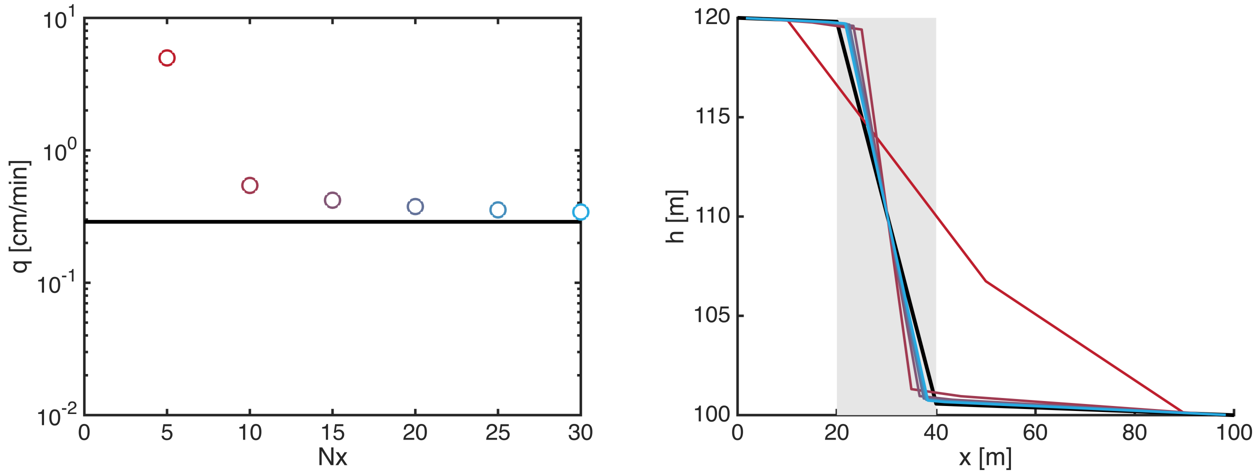
We will test the convergence of the numerical solution for both the head and the flux with different means as the grid is refined. We will consider the following grids

```
Nx = [1 2 3 4 5 6]*5;
Grid.xmin = 0; Grid.xmax = xa(end);
Kvec = K;
% create custom color map
N = length(Nx);
RED = repmat(col.red,N,1);
BLUE = repmat(col.blue,N,1);
ALPHA = repmat(linspace(0,1,N)',1,3);
COL = BLUE.*ALPHA + RED.*(1-ALPHA);
```

Arithmetic Mean

```
figure('position',[10 10 1200 600])
subplot 121
semilogy([0 Nx(end)],qa*[1 1]*6000,'k-','linewidth',3), hold on
xlim([0 Nx(end)]), ylim([1e-2 1e1])
xlabel 'Nx', ylabel 'q [cm/min]'
pbaspect([1 .8 1])

subplot 122
patch([20 40 40 20],[100 100 120 120],.9*[1 1 1],'edgecolor','none'), hold
on
plot(xa,Ta,'k-','linewidth',3)
xlabel 'x [m]', ylabel 'h [m]'
pbaspect([1 .8 1])

%% Compute numerical solution
i=1;
for n = Nx
    % build grid
    Grid.Nx = n;
    Grid = build_grid(Grid);

    % Thermal conductivity
    K = ones(Grid.Nx,1);
    K(Grid.xc<xa(4)) = Kvec(3);
    K(Grid.xc<xa(3)) = Kvec(2);
    K(Grid.xc<xa(2)) = Kvec(1);
```

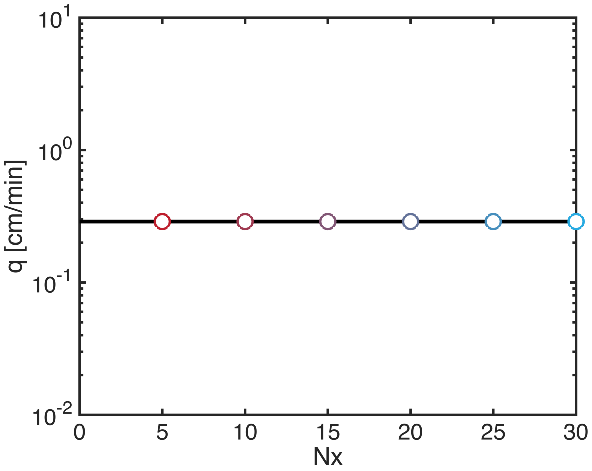
```matlab
    % Dirichlet BC's
    BC.dof_dir   = [Grid.dof_xmin;Grid.dof_xmax];
    BC.dof_f_dir = [Grid.dof_f_xmin;Grid.dof_f_xmax];
    BC.dof_neu   = [];
    BC.dof_f_neu = [];
    BC.g = interp1(xa,Ta,Grid.xc(BC.dof_dir));

    %% Discrete differential operators
    [D,G,C,I,M] = build_ops(Grid);

    % Heterogeneous coefficient
    Kd = spdiags(M*K,0,Grid.Nf,Grid.Nf);

    % Linear operator & r.h.s.
    L = -D*Kd*G; fs = spalloc(Grid.N,1,0);

    %% Build boundary operators
    [B,N,fn] = build_bnd(BC,Grid,I);

    %% Solve system
    u = solve_lbvp(L,fs+fn,B,BC.g,N);

    %% Compute fluxes
    q = comp_flux(D,Kd,G,u,fs,Grid,BC);

    subplot 121

semilogy(n,q(1)*6000,'o','markeredgecolor',COL(i,:),'markerfacecolor','w')
    xlim([0 Nx(end)])
    subplot 122
    plot(Grid.xc,u,'-','color',COL(i,:)), i=i+1;
    xlim([0,xa(end)])
    drawnow
end
```

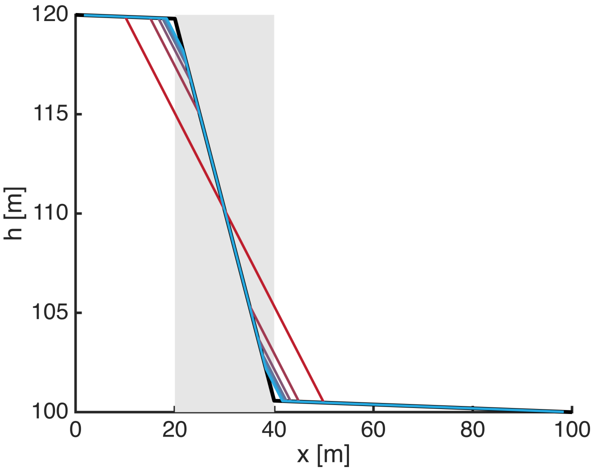


Harmonic Mean

```
figure('position',[10 10 1200 600])
subplot 121
semilogy([0 Nx(end)],qa*[1 1]*6000,'k-','linewidth',3), hold on
xlim([0 Nx(end)]), ylim([1e-2 1e1])
xlabel 'Nx', ylabel 'q [cm/min]'
pbaspect([1 .8 1])

subplot 122
patch([20 40 40 20],[100 100 120 120],.9*[1 1 1],'edgecolor','none'), hold
on
plot(xa,Ta,'k-','linewidth',3)
xlabel 'x [m]', ylabel 'h [m]'
pbaspect([1 .8 1])

%% Compute numerical solution
i=1;
for n = Nx
    % build grid
    Grid.Nx = n;
    Grid = build_grid(Grid);

    % Thermal conductivity
    K = ones(Grid.Nx,1);
    K(Grid.xc<xa(4)) = Kvec(3);
    K(Grid.xc<xa(3)) = Kvec(2);
    K(Grid.xc<xa(2)) = Kvec(1);
```

```
    % Dirichlet BC's
    BC.dof_dir   = [Grid.dof_xmin;Grid.dof_xmax];
    BC.dof_f_dir = [Grid.dof_f_xmin;Grid.dof_f_xmax];
    BC.dof_neu   = [];
    BC.dof_f_neu = [];
    BC.g = interp1(xa,Ta,Grid.xc(BC.dof_dir));

    %% Discrete differential operators
    [D,G,C,I,M] = build_ops(Grid);

    % Heterogeneous coefficient
    Kd = spdiags(1./(M*(1./K)),0,Grid.Nf,Grid.Nf);

    % Linear operator & r.h.s.
    L = -D*Kd*G; fs = spalloc(Grid.N,1,0);

    %% Build boundary operators
    [B,N,fn] = build_bnd(BC,Grid,I);

    %% Solve system
    u = solve_lbvp(L,fs+fn,B,BC.g,N);

    %% Compute fluxes
    q = comp_flux(D,Kd,G,u,fs,Grid,BC);

    subplot 121

semilogy(n,q(1)*6000,'o','markeredgecolor',COL(i,:),'markerfacecolor','w')
    xlim([0 Nx(end)])
    subplot 122
    plot(Grid.xc,u,'-','color',COL(i,:)), i=i+1;
    xlim([0,xa(end)])
    drawnow
end
```


q [cm/min]
10^{1}
10^{0}
10^{-1}
10^{-2}
0
5
10
15
20
25
30
Nx


h [m]
120
115
110
105
100
0
20
40
60
80
100
x [m]